A PROGRAMME FOR THE TAILORED SELECTION OF RESPONSE PATTERNS

Jeffrey Quentin Jackson

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A PROGRAMME FOR THE TAILORED SELECTION
OF RESPONSE PATTERNS

by

Jeffrey Quentin Jackson

June 1976

Thesis Advisor: R. A. Weitzman

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Programme for the Tailored Selection of Response Patterns		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jeffrey Quentin Jackson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1976
		13. NUMBER OF PAGES 34
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Tailored pattern analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The use of computer techniques to evaluate data in an attempt to find useful predictors of various criteria is of continuing interest. The use of stepwise pattern analysis to select predictors has shown promising results. This paper presents a refinement of this technique called TPAN, which allows the items selected to be "tailored" to the various

patterns of the previously selected items. This is followed by a discussion of the results obtained using TPAN to select a four-item pattern, from the responses to an advancement examination, that best predicts performance on the general classification test.

A PROGRAMME FOR THE TAILORED SELECTION OF RESPONSE PATTERNS

by

Jeffrey Quentin Jackson
Lieutenant-Commander, Canadian Forces
Bachelor of Science (Engineering Physics)

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL

June 1976

ABSTRACT

The use of computer techniques to evaluate data in an attempt to find useful predictors of various criteria is of continuing interest. The use of stepwise pattern analysis to select predictors has shown promising results. This paper presents a refinement of this technique called TPAN, which allows the items selected to be "tailored" to the various patterns of the previously selected items. This is followed by a discussion of the results obtained using TPAN to select a four-item pattern, from the responses to an advancement examination, that best predicts performance on the General Classification Test.

TABLE OF CONTENTS

I. INTRODUCTION.................................. 6

II. TECHNIQUE OF TAILORED PATTERN ANALYSIS........... 8

A. GENERAL.................................... 8

B. TPAN, A TAILORED PATTERN SELECTOR............ 9

III. AN APPLICATION OF TPAN........................ 14

IV. CONCLUSIONS AND RECOMMENDATIONS................. 17

Appendix A: LISTING OF ALGOL PROPROGRAMME TPAN......... 19

Appendix B: LISTING OF FORTRAN INPUT ROUTINE........... 31

I. INTRODUCTION

The increasing complexity of modern society has spawned a concurrent proliferation of specialized tasks which people are required to perform. As the training and skill necessary to carry out these tasks has increased, there has arisen a desire to select only those most likely to succeed to undergo such training and perform such tasks. Thus there is considerable interest in selection procedures and methods of prediction of success.

In the quest for better and better selectors for more and more specialized criteria, the complexity of testing procedures has grown. However, because of the costs of designing and administering large tests, methods are being sought to increase the validity of prediction from ever smaller sets of test items. The use of large digital computers and improved statistical techniques has aided this cause considerably.

One such technique to improve the predictive validity of a set of test items is called pattern analysis. Here, rather than aggregating the number of right and wrong answers into a single score, the pattern of right and wrong answers to the individual questions is analysed. The theoretical basis of this method is discussed by Lubin and Osborne in Reference 1, and Weitzman presents a summary of work on it through 1973 in Reference 2.

Folce [3] has developed pattern analysis into a computerized stepwise technique for selecting a subset of best predictors from a larger set of items. This programme

is called PAIN. The results of this programme compare very favourably tc the use of aggregate sccres. It is the intention of this paper to present a refinement cf this prccedure which would allow "tailcring" of the items selected so that the best item is selected fcr each pattern of responses tc the previously selected items.

II. TECHNIQUE OF PATTERN ANALYSIS

A. GENERAL

Stepwise pattern analysis is a technique employed to select, from a set of binary items, a small subset that is the best predictor for some criterion. These binary items may reflect correct or incorrect responses to test questions or indicate whether or not the subject is included in a demographic group, e.g., black or not black, age between 25 and 30 years or not. The criterion can also be binary, such as success or failure in training, or it may be continuous, e.g., final examination score.

Whether the criterion is continuous or binary, the process of item selection is the same. An item is selected and a pattern score is computed for all possible patterns using that item and previously selected items. The pattern score is obtained by computing the mean score on the criterion for all subjects having that pattern. For example, on the first item, the scores of all persons having an incorrect (zero) response on that item are averaged to give the zero pattern score, and the same for all persons having a correct (one) response. For the second item there are four possible patterns: correct on both items (11), incorrect on both items (00), correct on item one and incorrect on item two (10), and incorrect on item one and correct on item two (01). In all cases, the mean criterion score of the subjects in each category is assigned as that pattern score.

After the pattern scores are determined, each subject is assigned the pattern score appropriate to his pattern. The correlation between the subjects' pattern scores and their actual scores is then calculated. This calculation is repeated for each item in the set, and the item having the highest correlation coefficient is selected as the best item to be added to the subset.

Using this method a great deal of information can be obtained from relatively meager data. For instance, Folce was able to select only seven of the 70 items in the Electronics Technician Selection Test and obtain a correlation of better than 0.8 between the pattern score and the final grade assignment at the Electronics Technician School at San Diego, California. However, it should be possible to get even more information from the same sized subset by allowing different items to be selected for different subsets of the sample. That is, having selected the first item, the sample can be divided into two groups, those scoring a one on that item and those scoring a zero. It is quite possible that the next best predictor may be different for each of these groups, and different from the best predictor for the gruop as a whole. While PAIN selects the next item based on the whole group, tailored pattern analysis would allow a different item to be selected for each subgroup. A computer programme called TPAN has been developed to select such a tailored pattern of four items.

B. TPAN, A TAILORED PATTERN SELECTOR

TPAN is an ALGOL programme which will select a four-item pattern with the highest correlation between the pattern score for each individual and his actual score.

The programme first reads one card which must contain the number of binary test items for each subject in the sample (NITMS) and the number of subjects in the sample (NIS). The number of items is then passed to a FORTRAN subroutine called INPTTR to read in the data. This subroutine reads the complete data for one subject and passes back, to the main programme, the criterion score and an integer array of ones and zeros which are the item responses for that subject. Also, if it has reached the end of the file, the subroutine returns the actual number of records it has read so that the number of subjects (NIS) can be updated. To reduce the amount of memory required by the programme, TPAN compresses the data in the response array so that the responses to 32 items are contained in one word. Two new arrays are then formed, each having one entry for each subject in the sample. One array contains the criterion scores and the other the item responses. Each entry in the latter uses as many words as are required to contain the responses to all the items.

Most of the work is done by the subroutine BITPICKER. This routine, having been passed the array of scores and responses, selects the item from the responses that has the highest correlation between pattern scores and actual scores.

The subject's response to a particular item is determined by placing a one in a mask only in the bit corresponding to the item under consideration. A logical "and" operation is then performed with the word containing the subject's response. Only if his response to that one item was a one will the result of the "and" operation be other than zero. In such case his score will be added to the sum for the "one" responses and the number of "one" responses will be incremented. If a zero results from the "and", the changes will be made to the "zero" response data.

A running total is also kept of the sum of the squares of the scores of each individual.

When the responses of all the subjects have been checked the mean score for the zero and one responses is calculated, giving the pattern scores. These, along with the sum of the criterion scores and the sum of the squares of the criterion scores, are enough data to calculate the correlation coefficient. The computing formula for the Pearson product-moment correlation coefficient is used:

$$R = \frac{N \sum (cs)(ps) - (\sum cs)(\sum ps)}{\{[N \sum ps^2 - (\sum ps)^2][N \sum cs^2 - (\sum cs)^2]\}^{1/2}}$$

This procedure of obtaining pattern scores and then calculating the correlation coefficient is repeated for each item in the set. The item having the highest correlation coefficient is selected.

To facilitate the use of this routine for iterations when it is desired only to use a subset of subjects who had a particular pattern, a pointing vector is used rather than directly using the arrays of scores and responses. That is: the subroutine BITPICKER is always passed the total array of responses and scores. It is also passed another array containing the positions in the main array of all subjects who are to be used in the calculation. This is the so called pointing vector.

For example, to select the first item the pointing vector contains all the integers up to and including the total number of subjects in the sample. Thus when the subroutine checks each subject whose number is in the pointing vector, it checks the whole set. However, as the data for each subject is checked, his position number is put into one of two vectors depending on whether the response to that item is a one or a zero. These two vectors, for the item with the highest correlation coefficient, are passed back to the main programme.

When the second item is to be picked, BITPICKER is first passed the pointing vector to those subjects having a zero response to the first item. The subroutine will then pick the item having the highest correlation only for those subjects having the zero response. The subroutine is then called again but with the pointing vector for the one responses. Thus, a different second item may be picked for this subgroup. In each case two new pointing vectors are passed back to the main programme, pointing to those subjects having a zero and those having a one response to the chosen second items.

The main body of the programme is, therefore, a series of calls to the subroutine BITPICKER passing it the arrays of scores (SCORE) and responses (RESP), the appropriate pointing vector (PTR for the first item), and the number of entries in that vector (NIS). The subroutine returns two new pointing vectors (PTR0 and PTR1 for the first item), as well as the correlation coefficient (R), item number (ITM), totals of ones (TOT1) and zeros (TOT0), and the pattern scores for ones (MPS1) and zeros (MPS0). There are also masks passed back and forth to indicate which items have already been chosen (MASKIN and MASK) and standard accounting data of the total number of items (NITMS) and the number of words required to hold all the items at 32 items per word (NSEGS).

On the second call, the best item for those subjects having the zero response to item one is desired. Therefore, the data passed are the pointing vector PTR0 and its length TOT0. The data returned are: correlation coefficient R0, item number ITM0, and the pointing vectors, totals and pattern scores, PTR01, TOT01, MPS01, PTR00, TOT00, and MPS00.

The final result of TPAN is a set of 16 patterns described by the binary numbers 0000 through 1111. Each binary digit represents the response to one of the four items selected. The first item will be the same for all

patterns. There may be two different second items (one for each response to item one), four third items and eight fourth items. The final step in the programme is to calculate an overall correlation coefficient. Each subject is assigned the pattern score appropriate to his responses on the selected items. The correlation coefficient is then calculated using the same algorithm as for the individual items.

An additional facility provided by TPAN is the ability to set bounds on the criterion scores which it is desired to use. This is done by including two more numbers on the single ALGOL input data card; these are the upper limit of the desired scores and the lower limit. As the data records are read, each score is checked against these bounds and if it is outside the limits that record is rejected. The number of the record is printed out, as well as the score on which it was rejected. After all the records have been read, the number in the sample is revised to allow for the rejected records.

A complete listing of the ALGOL programme is contained in appendix A.

III. AN APPLICATION OF TPAN

In order to test the programme, TPAN was run using as data the results of an advancement examination to pay grade 7 for boiler technicians. The source data contained the results for approximately 1100 enlisted men for the 150 items on this examination. From these responses plus an additional item indicating whether the race of the individual was black, TPAN was to select the four best items to predict the subject's score on the General Classification Test (GCT).

A valid range of 1 to 99 was set for the GCT scores and a number of records were outside this range (the field contained either a zero or non-numeric data). TPAN eliminated these records and the final sample contained 1024 subjects. The results obtained from this run are given in table 1.

The value of the correlation coefficients given in the table are those used in selection of the items and, hence, represent the correlation only within the subset of subjects having the pattern shown for the previously selected items. It will be noted that these correlations are all rather small, ranging from 0.16 to 0.50. This is to be expected, however, as the advancement examination is not intended to measure the same qualities as the GCT. This is further borne out by the fact that the first item chosen, that is, the single best indicator of performance on the GCT among the items considered, was item 1, race.

TABLE 1

SELECTION OF ITEMS FROM ADVANCEMENT EXAMINATION AS PREDICTORS FOR GCT

pattern	correl- ation	items	number of 0's	mean score	number of 1's	mean score
0,1	.247	1	896	47.60	128	41.73
00,01	.242	1,24	399	45.56	497	49.23
000,001	.200	1,24,7	192	44.08	207	46.93
0000,0001	.217	1,24,7,104	110	42.75	82	45.87
0010,0011	.256	1,24,7,23	124	45.48	83	49.10
010,011	.218	1,24,104	217	47.39	280	50.66
0100,0101	.164	1,24,104,93	153	46.65	64	49.16
0110,0111	.222	1,24,104,86	124	48.80	156	52.15
10,11	.348	1,7	53	38.45	75	44.05
100,101	.362	1,7,104	32	36.09	21	42.05
1000,1001	.453	1,7,104,23	27	34.67	5	43.80
1010,1011	.495	1,7,104,13	4	50.00	17	40.17
110,111	.341	1,7,120	27	40.89	48	45.83
1100,1101	.450	1,7,120,101	14	43.71	13	37.85
1110,1111	.321	1,7,120,55	16	48.81	32	44.34

Even given these less than ideal circumstances, the overall correlation for the four-item patterns was 0.47. This compares favourably with the figure of 0.40 obtained for four items selected by PAIN. Moreover, one of the disadvantages of PAIN is the amount of time and computer memory required to run it. For the 1100 subject sample PAIN required 400,000 bytes of memory and 4 minutes to run. TPAN on the other hand required only 180,000 bytes and ran in slightly over 3 minutes. This is partly because of the fact that only two patterns are assessed on each iteration and partly due to the more efficient handling of the algorithm allowed by AIGCL.

IV. CONCLUSIONS AND RECOMMENDATIONS

The results of the study indicate that tailoring the item selection in pattern analysis enables more information to be extracted from a four-item pattern than if straight stepwise selection based on the whole group is used. However, the actual advantage gained in terms of the amount of predication per test item is questionable. There are, in fact, eight sets of four test items using a total of up to 15 different items. Therefore, if TPAN were to be used to select items to be included in a minimum length test, its performance would have to be compared to PAIN selecting a 15 item subset. On the other hand, if it is desired to look at existing data in an attempt to predict some output, TPAN should present a distinct advantage.

There are several areas where TPAN could be improved and extended. The first is the data printed out. As mentioned, the correlation coefficients that are given are those within the subset used to pick the next item. More useful values would be the overall correlations at the end of the selection of all second, third, and fourth items. The final one is the only one calculated at present. To accomplish this would require only the the accumulation of one or two more items of data, which are already available, and two additional correlation calculations.

Another shortcoming of the programme is its response when it reaches a point of indifference to all items, i.e., the correlation coefficients for all items is zero. At present, in this situation, the programme prints an obviously erroneous item number (-32), and sets all of the statistics (mean scores and totals of zero and one responses) to zero. This action will disrupt the calculation of the overall correlation coefficient. The most

reasonable corrective action in this case would be to terminate selection of items and, when calcutating the overall correlation coefficient, use the pattern and pattern scores derived for the last good item.

Increasing the number of items in the pattern presents no programming problems. It is simply necessary to add more calls to the subroutine BITPICKER, passing it the appropriate pointing vector. The problems encountered are statistical. The numbers of patterns and possible different items doubles with each addition of one item to the pattern. With 1100 subjects in the sample, there are already some subgroups of less than 20 subjects. The validity of items selected on the basis of such small samples is questionable.

A final and very interesting area for increasing the scope of the programme would be to include some ability to manipulate continuous data as well as binary items. The programme could be changed to determine the correlation between any pair of continuous attributes of the subgroups having pattern responses selected by TPAN. All that would be required would be to read in an array or arrays of the values of the continuously variable data for each subject. Then, after each item was selected, the pointing vector produced by the BITPICKER subroutine could be used to select the appropriate subjects' data from the arrays of continuous variables. Each correlation coefficient thus derived would be for a subgroup having a particular pattern. Such a routine could be used to determine for which of several subgroups, having different patterns of responses, the correlation was highest. Such a programme could also answer other interesting questions. For example, if we select a subgroup having a pattern with high correlation between pattern and actual scores, how does it affect the correlation between an independent continuous variable and the same criterion score?

APPENDIX A

LISTING OF ALGOL PROGRAMME TRAN

The following pages contain a listing of the source file of the ALGOL programme. The first two columns would not be part of an input deck, but are included to facilitate reading the programme. A number in the first column indicates when a block of code starts. The same number in the next column indicates the end of that block.

```
1-      BEGIN
--     COMMENT THIS PROGRAMME WILL SELECT FOUR ITEMS
--     TAILORED TO THE PREVIOUS PATTERNS, WHICH GIVE THE HIGHEST
--     CORRELATION BETWEEN THE PATTERN SCORES AND THE ACTUAL
--     SCORE FOR EACH SUBJECT.
--
--     TO USE THIS PROGRAMME A FORTRAN SUBROUTINE IS REQUIRED WHICH
--     WILL READ THE INPUT DATA FOR ONE SUBJECT AND RETURN THE
--     CRITERION SCORE AND A VECTOR OF ONES AND ZEROS WHICH
--     CONTAINS THE RESPONSES OF THAT SUBJECT TO ALL ITEMS.
--     THE SUBROUTINE WILL BE PASSED THE NUMBER OF ITEMS TO BE
--     EXPECTED WHICH MUST BE THE LENGTH OF THE RESPONSE VECTOR.
--     IN ADDITION IF THE END OF FILE IS REACHED, THE SUBROUTINE
--     MUST CHANGE THE VALUE OF A FLAG FROM ZERO TO THE NUMBER OF
--     RECORDS ACTUALLY READ.
--     THE TITLE OF THE SUBROUTINE MUST BE INPTTR AND THE ORDER
--     OF THE PARAMETERS PASSED MUST BE RESPONSE ARRAY, SCORE,
--     NUMBER OF ITEMS, AND THE NUMBER OF RECORDS READ.  THE LAST
--     MUST BE ZERO UNLESS THE END OF THE INPUT FILE IS REACHED.
--
--     THERE ALSO MUST BE ONE INPUT DATA CARD FOR THE ALGOL
--     PROGRAMME.  IT MUST CONTAIN THE FOLLOWING DATA IN THIS ORDER.
--     THE NUMBER OF ITEMS FOR EACH SUBJECT, THE NUMBER OF SUBJECTS,
--     THE UPPER BOUND ON ACCEPTABLE SCORES, AND THE LOWER BOUND
--     ON THE ACCEPTABLE SCORES.  THESE LAST TWO ITEMS MUST BE GIVEN
--     VALUES OR ELSE THE PROGRAMME WILL REJECT ALL RECORDS.;
--


--
--     PROCEDURE INPTTR(INTEGER ARRAY RSP(*);REAL RESULT SCR;
--     INTEGER VALUE NITMS;INTEGER VALUE RESULT NRC); FORTRAN "INPTTR";
```

```
--      PROCEDURE BITPICKER (INTEGER ARRAY PTR,PSV0,PSV1(*);
--          BITS ARRAY MASKIN,MASKOUT(*);INTEGER VALUE NIS;
--         INTEGER RESULT ONTOT,OFTOT,ITMNO;REAL RESULT RSVD,MPS0,MPS1,
--          WSVD;INTEGER VALUE NSEGS,NITMS;
--          REAL ARRAY SCORE(*); BITS ARRAY RESP(*,*));
2-      BEGIN
--       INTEGER ARRAY PTR0,PTR1(1::NIS+1);
--       REAL OFFSUM,ONSUM,OFFMEAN,ONMEAN,OFFREC,ONFREC;
--       INTEGER SEGSVD, SEG,SHFTSVD,SHFTCTR,P0,P1,N,J;
--       REAL SUMPS,SUMPS2,W,SUMCS1,SUMCS0,R1,R2,R3,SUMCS,SUMCS2;
--       BITS MASK, MASKSVD, BT;
--       SEGSVD := SHFTSVD := 0;
--       WSVD := RSVD := MPS0 := MPS1 := 0.;
--       SHFTCTR := -1;
--       SEG := OFTOT := ONTOT := 0;
--
--       NXTITM:
--                SHFTCTR:= SHFTCTR+1;
--           IF (32*(SEG-1)+SHFTCTR) > NITMS THEN GO TO ALITMSCHKD;
--           IF (SHFTCTR > 32) OR (SHFTCTR = 0) THEN
3-                BEGIN
--                MASK := #1;
--                SEG := SEG + 1;
--                SHFTCTR:= 1;
-3                END
--                ELSE
--                MASK := MASK SHL 1;
--           BT := MASK AND MASKIN(SEG);
--           IF BT ¬= #0 THEN GO TO NXTITM;
--       COMMENT WITH MASK SET TO SELECT A PARTICULAR ITEM
--       SUM THE CRITERION SCORES FOR ALL SUBJECTS SCORING
--       ZERO FOR THAT ITEM AND SCORING ONE FOR THAT ITEM.
--       ALSO A VECTOR IS CONSTRUCTED TO POINT TO EACH
--       SUBJECT HAVING A ZERO RESPONSE AND A ONE RESPONSE;
--       P0 := P1 := 1;
--       SUMCS2 := 0.;
--       OFFREC := ONFREC := OFFSUM := ONSUM := 0;
```

```
--      FOR J := 1 UNTIL NIS +1  DO
--          PTRO(J) := PTR1(J) := 0;
--          FOR K:=  1 UNTIL NIS DO
3-              BEGIN COMMENT FREQUENCY CALCULATION;
--              J := PTR(K);
--              BT := MASK AND RESP(J,SEG);
--              IF BT =    #0 THEN
4-                  BEGIN COMMENT ZERO RESPONSE TALLY;
--                  OFFSUM:= OFFSUM + SCORE(J);
--                  SUMCS2 := SUMCS2 + SCORE(J)**2;
--                  PTRO(PO) := J;
--                  PO:= PO + 1;
--                  OFFREC := OFFREC + 1;
-4                  END COMMENT ZERO RESPONSE TALLY;
--                  ELSE
4-                  BEGIN COMMENT 1 RESPONSE TALLY;
--                  ONSUM:= ONSUM + SCORE(J);
--                  PTR1(P1) := J;  COMMENT STORE SUBJECT NUMBER
--                                  IN 1 RESPONSE POINTER;
--                  SUMCS2 := SUMCS2 + SCORE(J)**2;
--                  P1 := P1 + 1;
--                  ONFREC := ONFREC + 1 ;
-4                  END; COMMENT 1 RESPONSE TALLY;
-3              END; COMMENT FREQUENCY CALCULATION;
--              COMMENT  CALCULATE MEANS;
--          IF OFFREQ ¬= 0 THEN
--              OFFMEAN := OFFSUM/OFFREC
--                  ELSE
--              OFFMEAN := 0;
--          IF ONFREQ ¬= 0 THEN
--              ONMEAN := ONSUM/ONFREC
--                  ELSE
--              ONMEAN := 0;
3-          BEGIN;
```

```
--      CCMMENT  CALCULATE CORRELATION COEFFICIENT R2;
--          SUMPS :=OFFSUM + CNSUM;
--          SUMPS2 := OFFREQ*OFFMEAN*CFFMEAN + CNFREQ*ONMEAN*ONMEAN;
--          N := 1;
--          SUMCS := SUMPS;
--          SUMCSO := OFFSUM;
--          SUMCS1 := ONSUM;
--
--          W := SUMCS1 * ONMEAN + SUMCSO * OFFMEAN;
--          R1 := (NIS*SUMPS2) - (SUMPS*SUMPS);
--          IF ABS(R1) < 1.'-05 THEN R2 := 0 ELSE
4-              BEGIN
--              R3 := R1*(NIS*SUMCS2 - SUMCS**2);
--              IF R3 <1.'-05 THEN R2 := 0. ELSE
--              R2 := (NIS*W - SUMCS*SUMPS)/SQRT(R3);
-4              END
-3          END; COMMENT CORRELATION COEFICIENT CALCULATION;

--          IF ABS(R2) > ABS(RSVD) THEN
3-              BEGIN  COMMENT  SAVE HIGHEST CORRELATION COEFICIENT, SELECTOR
--                  MASK, AND SEGMENT NUMBER;
--              RSVD := R2;
--              WSVD := W;
--              MPSO := OFFMEAN;
--              MPS1 := CNMEAN;
--              MASKOUT(SEG) := MASKIN(SEG) OF MASK;
--              SEGSVD := SEG;
--              FOR J := 1 UNTIL PO DC PSVO(J) := PTRO(J);
--              FOR J := 1 UNTIL P1 DC PSV1(J) := PTR1(J);
--              OFTOT := PO - 1;
--              ONTOT := P1 - 1;
--              MASKSVD := MASK;
--              SHFTSVD := SHFTCTR;
-3              END;
--          GC TO NXTITM;
--
--      ALITMSCHKD:
3-      BEGIN
--      ITMNO := (SEGSVD - 1)*32 + SHFTSVD;
-3      END;
-2      END;
```

```
2-    BEGIN   COMMENT  MAIN PROGRAMME;
--    REAL SUMCS,SUMCS2,R,MPSO,MPS1,SUMPS,SUMPS2,W,WINC,
--    R1,R2,R3,BSCR, LWR, UPR;
--    INTEGER NITMS, NIS, SEG, NSEGS, INCT,TOTO, TCT1, ITM, NRD;
3-        BEGIN
--        READ (NITMS,NIS,UPR,LWR);
--        NSEGS := (NITMS + 31) DIV 32;
-3        END;
3-    BEGIN
--    REAL ARRAY SCORE(1::NIS);
--    BITS ARRAY RESP(1::NIS,1::NSEGS);
--    BITS ARRAY MASKIN, MASK(1::NSEGS);
--    INTEGER ARRAY PTRO,PTR1,PTR(1::NIS+1);
4-    BEGIN
--    BITS ACCUM,MSK;
--    INTEGER ARRAY RSP(1::NITMS);
--    INCT := NRD := 0;
--    SUMCS := SUMCS2 := 0.;
--    FOR J := 1 UNTIL NIS  DO
5-    BEGIN
--    AGAIN:
--    INPTTR (RSP, SCORE(J), NITMS, NRD);
--    IF NRD¬=0 THEN GO TO SCRDUP;
6-    IF (SCORE(J) < LWR) OR (SCORE(J) > UPR) THEN BEGIN
--    WRITE ("CARD NUMBER",  J ," SCORE",SCORE(J));
-6    GO TO AGAIN; END;
--    INCT := INCT + 1;
--    SUMCS := SUMCS + SCORE(J);
--    SUMCS2 := SUMCS2 + SCORE(J)**2;
--    MSK := #1;
--    SEG := 1;
--    ACCUM := #0;
--    FOR K := 1 UNTIL NITMS DO
6-         BEGIN
--         IF RSP (K) = 1 THEN
--             ACCUM := ACCUM OR MSK;
--         MSK := MSK SHL 1;
--         IF MSK = #0 THEN
7-             BEGIN
--             RESP(J,SEG) := ACCUM;
--             MSK := #1;
--             SEG := SEG + 1;
--             ACCUM := #0;
-7             END;
-6        END;
-5        END;
--        GO TO NTSCRDUP;
```

```
--      SCRCUP:
5-           BEGIN
--         WRITE  ("ACTUAL NUMBER OF RECORDS REAC IS",NRC,
--                     "NUMBER TC BE EXPECTED WAS",NIS);
--         WRITE ( "  ");
--         WRITE (" ");
-5           END;
--
--      NTSCRCUP:
--          NIS := INCT;
--          BSCR := SUMCS/NIS;
--          WRITE ("THE BASE SCORE IS",BSCR," FOR",NIS," SUBJECTS");
--           WRITE ( "  ");
--           WRITE ( "  ");
-4          ENC;
--       FOR J:=1 UNTIL NIS DO PTR(J):=J;
--      FCR J := 1 UNTIL NSEGS DO MASKIN(J) := MASK(J) := #0;
--      WRITE ("PATTERN       CORRELATICN           ITEM     NC. OF ZERCS
--         MPS ZERO       NO. OF ONES        MPS CNES");
--      WRITE (" ");
--      BITPICKER (PTR,PTRO,PTR1,MASKIN,MASK,NIS,TCT1,TOTO,
--      ITM,R,MPSO,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--      WRITE ("NONE        ",R,ITM,TCTO,MPSO,TOT1,"       ",MPS1);
--
4-        BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ZERC RESPONSE TO
--          FIRST ITEM WILL NCW BE SELECTEC;
--        INTEGER ARRAY PTROO,PTRO1(1::TCTO+1);
--        BITS ARRAY MASKO(1::NSEGS);
--        REAL RO;
--        INTEGER  ITMO, TCTOO, TOTO1;
--        FOR J := 1 UNTIL TCTO+1 DO
--          PTROO(J) := PTRO1(J) := 0;
--        FOR J := 1 UNTIL NSEGS DO
--          MASKO (J) := #0;
--        BITPICKER (PTRO,PTROO,PTRO1,MASK,MASKO,TCTO,TOTO1,TOTOO,
--        ITMO,RO,
--        MPSO,MPS1,WINC,NSEGS,NITMS,SCCRE,RESP);
--        WRITE ("  O       ",RO, ITMO,TCTOO,MPSO,TOTO1,"      ",MPS1);
--
5-          BEGIN COMMENT THE TAILCREC SUBSET OF ITEMS FOR THE ZERO RESPONSE
--            TO PATTERN O WILL NCW BE SELECTEC;
--          INTEGER ARRAY PTROOO,PTROO1(1::TOTOO+1);
--          BITS ARRAY MASKOO(1::NSEGS);
--          REAL ROO;
--          INTEGER ITMOO,TOTOOO,TCTOO1;
--          FOR J := 1 UNTIL TOTOO+1 DC
--            PTROOO(J) := PTROO1(J) := 0;
```

```
--          FOR J := 1 UNTIL NSEGS DO
--            MASKOO (J) := #0;
--          BITPICKER (PTROO,PTROOO,PTROO1,MASKO,MASKOO,TCTOO,TOTOO1,TOTOOO,
--          ITMOO,ROO,
--          MPSO,MPS1,WINC,NSEGS,NITMS,SCCRE,RESP);
--          WRITE ("    OO    ",ROO, ITMOO,TOTOOO,MPSO,TOTOO1,"      ",MPS1);
--
6-              BEGIN COMMENT THE TAILOREC SUBSET CF ITEMS FOR THE ZERO RESPONS
--              TO PATTERN OO WILL NOW BE SELECTEC;
--              INTEGER ARRAY PTROOOO,PTROOO1(1::TCTOOO+1);
--              BITS ARRAY MASKOOO(1::NSEGS);
--              REAL ROOO;
--              INTEGER ITMOOO,TOTOOOO,TCTOOO1;
--              FOR J := 1 UNTIL TCTOOO+1 CO
--                PTROOOO(J) := PTROOO1(J) := 0;
--              FOR J := 1 UNTIL NSEGS CC
--                MASKOOO (J) := #0;
--              BITPICKER (PTROOO,PTROOOO,PTROOO1,MASKOO,MASKOOO,TOTOOO,TOTOOO1
--              TCTOOOO,ITMOOO,ROOO,
--              MPSO,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--           WRITE ("      OOO ",ROOO,ITMOOO,TOTOOOO,MPSO,TCTOOO1,"      ",MPS1)
--              SUMPS := MPSO*TOTOOOO + MPS1*TCTOOO1;
--              SUMPS2 := (TOTOOOO*MPSO*MPSO) + (TCTOOO1*MPS1*MPS1);
--              W := WINC;
-6              END;
--
6-              BEGIN COMMENT THE TAILOREC SUBSET CF ITEMS FOR THE CNE RESPONSE
--                TO PATTERN OO WILL NOW BE SELECTEC;
--              INTEGER ARRAY PTROO10,PTROO11(1::TCTOO1+1);
--              BITS ARRAY MASKOO1(1::NSEGS);
--              REAL ROO1;
--              INTEGER ITMOO1,TOTOO10,TCTOO11;
--              FOR J := 1 UNTIL TCTOO1+1 DO
--               PTROO10(J) := PTROO11(J) := 0;
--              FOR J := 1 UNTIL NSEGS CC
--               MASKOO1 (J) := #0;
--              BITPICKER (PTROO1,PTROO10,PTROO11,MASKOO,MASKOO1,TOTOO1,TOTOO11
--              TOTOO10,ITMOO1,ROO1,
--              MPSO,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--            WRITE ("      OO1 ",ROO1,ITMOO1,TOTOO10,MPSO,TOTOO11,"      ",MPS1);
--                WRITE ("    ");
--              W := W + WINC;
--              SUMPS := SUMPS + (TOTOO10*MPSO) + (TCTOO11*MPS1);
--              SUMPS2 := SUMPS2 + (TOTOO10*MPSO*MPSO) + (TCTOO11*MPS1*MPS1);
-6                END;
-5          ENC;
```

```
5-          BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ONE RESPONSE
--                TO ITEM 0 WILL NOW BE SELECTED;
--          INTEGER ARRAY PTR010,PTR011(1::TOT01+1);
--          BITS ARRAY MASK01(1::NSEGS);
--          REAL R01;
--          INTEGER ITM01,TOT010,TOT011;
--          FOR J := 1 UNTIL TOT01+1 DO
--              PTR010(J) := PTR011(J) := 0;
--          FOR J := 1 UNTIL NSEGS DO
--              MASK01 (J) := #0;
--          BITPICKER (PTR01,PTR010,PTR011,MASK0,MASK01,TOT01,TOT011,
--          TOT010,ITM01,R01,
--          MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--          WRITE ("    01    ",R01,ITM01,TOT010,MPS0,TOT011,"      ",MPS1);
--
6-            BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ZERO RESPONSE
--                  TO ITEM 01 WILL NOW BE SELECTED;
--            INTEGER ARRAY PTR0100,PTR0101(1::TOT010+1);
--            BITS ARRAY MASK010(1::NSEGS);
--            REAL R010;
--            INTEGER ITM010,TOT0100,TOT0101;
--            FOR J := 1 UNTIL TOT010+1 DO
--                PTR0100(J) := PTR0101(J) := 0;
--            FOR J := 1 UNTIL NSEGS DO
--                MASK010 (J) := #0;
--            BITPICKER (PTR010,PTR0100,PTR0101,MASK01,MASK010,TOT010,TOT0101,
--            TOT0100,ITM010,R010,
--            MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--           WRITE ("      010 ",R010,ITM010,TOT0100,MPS0,TOT0101,"      ",MPS1);
--            W := W +  WINC;
--            SUMPS := SUMPS + TOT0100*MPS0 + TOT0101*MPS1;
--            SUMPS2 := SUMPS2 + TOT0100*MPS0*MPS0 + TOT0101*MPS1*MPS1;
-6            END;
--
6-            BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ONE RESPONSE
--                  TO ITEM 01 WILL NOW BE SELECTED;
--            INTEGER ARRAY PTR0110,PTR0111(1::TOT011+1);
--            BITS ARRAY MASK011(1::NSEGS);
--            REAL R011;
--            INTEGER ITM011,TOT0110,TOT0111;
--            FOR J := 1 UNTIL TOT011+1 DO
--                PTR0110(J) := PTR0111(J) := 0;
--            FOR J := 1 UNTIL NSEGS DO
--                MASK011 (J) := #0;
--            BITPICKER (PTR011,PTR0110,PTR0111,MASK01,MASK011,TOT011,TOT0111,
--            TOT0110,ITM011,R011,
--            MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--            WRITE ("      011 ",R011,ITM011,TOT0110,MPS0,TOT0111,"      ",MPS1);
```

```
--            WRITE ("    ");
--            W := W + WINC;
--            SUMPS := SUMPS + TOT0110*MPS0 + TOT0111*MPS1;
--            SUMPS2 := SUMPS2 + TOT0110*MPS0*MPS0 + TOT0111*MPS1*MPS1;
-6            END;
-5          END;
-4        END;
--
4-        BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ONE RESPONSE
--              TO THE FIRST ITEM WILL NOW BE SELECTED;
--        INTEGER ARRAY PTR10,PTR11(1::TOT1+1);
--        BITS ARRAY MASK1(1::NSEGS);
--        REAL R1;
--        INTEGER ITM1,TOT10,TOT11;
--        FOR J := 1 UNTIL TOT1+1 DO
--            PTR10(J) := PTR11(J) := 0;
--        FOR J := 1 UNTIL NSEGS DO
--            MASK1 (J) := #0;
--        BITPICKER (PTR1,PTR10,PTR11,MASK,MASK1,TOT1,TOT11,
--        TOT10,ITM1,R1,
--        MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--        WRITE ("  1        ",R1,ITM1,TOT10,MPS0,TOT11,"     ",MPS1);
5-          BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ZERO RESPONSE
--                TO PATTERN 1 WILL NOW BE SELECTED;
--          INTEGER ARRAY PTR100,PTR101(1::TOT10+1);
--          BITS ARRAY MASK10(1::NSEGS);
--          REAL R10;
--          INTEGER ITM10,TOT100,TOT101;
--          FOR J := 1 UNTIL TOT10+1 DO
--              PTR100(J) := PTR101(J) := 0;
--          FOR J := 1 UNTIL NSEGS DO
--              MASK10 (J) := #0;
--          BITPICKER (PTR10,PTR100,PTR101,MASK1,MASK10,TOT10,TOT101,
--          TOT100,ITM10,R10,
--          MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--          WRITE ("    10    ",R10,ITM10,TOT100,MPS0,TOT101,"      ",MPS1);
6-            BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ZERO RESPONSE
--                  TO PATTERN 10 WILL NOW BE SELECTED;
--            INTEGER ARRAY PTR1000,PTR1001(1::TOT100+1);
--            BITS ARRAY MASK100(1::NSEGS);
--            REAL R100;
--            INTEGER ITM100,TOT1000,TOT1001;
--            FOR J := 1 UNTIL TOT100+1 DO
--                PTR1000(J) := PTR1001(J) := 0;
--            FOR J := 1 UNTIL NSEGS DO
--                MASK100 (J) := #0;
```

```
--          BITPICKER (PTR100,PTR1000,PTR1001,MASK10,MASK100,TOT100,TOT1001,
--          TOT1000,ITM100,R100,
--          MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--         WRITE ("      100 ",R100,ITM100,TOT1000,MPS0,TOT1001,"      ",MPS1);
--          W := W + WINC;
--          SUMPS := SUMPS + TOT1000*MPS0 + TOT1001*MPS1;
--          SUMPS2 := SUMPS2 + TOT1000*MPS0*MPS0 + TOT1001*MPS1*MPS1;
-6          END;
6-          BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ONE RESPONSE
--                TO PATTERN 10 WILL NOW BE SELECTED;
--          INTEGER ARRAY PTR1010,PTR1011(1::TOT101+1);
--          BITS ARRAY MASK101(1::NSEGS);
--          REAL R101;
--          INTEGER ITM101,TOT1010,TOT1011;
--          FOR J := 1 UNTIL TOT101+1 DO
--              PTR1010(J) := PTR1011(J) := 0;
--          FOR J := 1 UNTIL NSEGS DO
--              MASK101 (J) := #0;
--          BITPICKER (PTR101,PTR1010,PTR1011,MASK10,MASK101,TOT101,TOT1011,
--          TOT1010,ITM101,R101,
--          MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--          WRITE ("      101 ",R101,ITM101,TOT1010,MPS0,TOT1011,"      ",MPS1);
--          WRITE ("   ");
--          W := W + WINC;
--          SUMPS := SUMPS + TOT1010*MPS0 + TOT1011*MPS1;
--          SUMPS2 := SUMPS2 + TOT1010*MPS0*MPS0 + TOT1011*MPS1*MPS1;
-6          END;
-5        END;
5-        BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ONE RESPONSE
--              TO PATTERN 1 WILL NOW BE SELECTED;
--        INTEGER ARRAY PTR110,PTR111(1::TOT11+1);
--        BITS ARRAY MASK11(1::NSEGS);
--        REAL R11;
--        INTEGER ITM11,TOT110,TOT111;
--        FOR J := 1 UNTIL TOT11+1 DO
--            PTR110(J) := PTR111(J) := 0;
--        FOR J := 1 UNTIL NSEGS DO
--            MASK11 (J) := #0;
--        BITPICKER (PTR11,PTR110,PTR111,MASK1,MASK11,TOT11,TOT111,
--        TOT110,ITM11,R11,
--        MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--        WRITE ("   11     ",R11,ITM11,TOT110,MPS0,TOT111,"      ",MPS1);
6-          BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ZERO RESPONSE
--                TO PATTERN 11 WILL NOW BE SELECTED;
--          INTEGER ARRAY PTR1100,PTR1101(1::TOT110+1);
--          BITS ARRAY MASK110(1::NSEGS);
--          REAL R110;
--          INTEGER ITM110,TOT1100,TOT1101;
```

```
--            FOR J := 1 UNTIL TOT110+1 DO
--                 PTR1100(J) := PTR1101(J) := 0;
--            FOR J := 1 UNTIL NSEGS DO
--                 MASK110 (J) := #0;
--            BITPICKER (PTR110,PTR1100,PTR1101,MASK11,MASK110,TOT110,TOT1101,
--            TOT1100,ITM110,R110,
--            MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--          WRITE ("      110 ",R110,ITM110,TOT1100,MPS0,TOT1101,"      ",MPS1);
--            W := W + WINC;
--            SUMPS := SUMPS + TOT1100*MPS0 + TOT1101*MPS1;
--            SUMPS2 := SUMPS2 + TOT1100*MPS0*MPS0 + TOT1101*MPS1*MPS1;
-6            END;
6-            BEGIN COMMENT THE TAILORED SUBSET OF ITEMS FOR THE ONE RESPONSE
--                  TO PATTERN 11 WILL NOW BE SELECTED;
--            INTEGER ARRAY PTR1110,PTR1111(1::TOT111+1);
--            BITS ARRAY MASK111(1::NSEGS);
--            REAL R111;
--            INTEGER ITM111,TOT1110,TOT1111;
--            FOR J := 1 UNTIL TOT111+1 DO
--                 PTR1110(J) := PTR1111(J) := 0;
--            FOR J := 1 UNTIL NSEGS DO
--                 MASK111 (J) := #0;
--            BITPICKER (PTR111,PTR1110,PTR1111,MASK11,MASK111,TOT111,TOT1111,
--            TOT1110,ITM111,R111,
--            MPS0,MPS1,WINC,NSEGS,NITMS,SCORE,RESP);
--          WRITE ("      111 ",R111,ITM111,TOT1110,MPS0,TOT1111,"      ",MPS1);
--            WRITE ("   ");
--            WRITE ("   ");
--            W := W + WINC;
--            SUMPS := SUMPS + TOT1110*MPS0 + TOT1111*MPS1;
--            SUMPS2 := SUMPS2 +TOT1110*MPS0*MPS0 + TOT1111*MPS1*MPS1;
-6            END;
-5         END;
-4      END;
--      R1 := (NIS*SUMPS2) - (SUMPS*SUMPS);
--      IF ABS(R1) < 1.'-05 THEN R2 := 0 ELSE
4-       BEGIN
--       R3 := R1*(NIS*SUMCS2 - SUMCS**2);
--       IF R3 < 1.'-05 THEN R2 := 0. ELSE
--          R2 := (NIS*W - SUMCS*SUMPS)/SQRT(R3);
-4       END;
--     WRITE ("THE OVERALL CORRELATION IS",R2);
-3    END;
-2    END;
-1    END.
```

APPENDIX B

LISTING OF FORTRAN INPUT ROUTINE

The following listing is the FORTRAN input routine used with data supplied on the advancement examination. The file was 160 characters long, with race in column 6 followed by the responses on the 150 questions. The GCT score was in columns 157 and 158. The data was read into an integer array and passed back to the main programme.

```
      SUBROUTINE INPTTR (I,SCORE,NITMS,NRD)
       DIMENSION I (NITMS)
      DATA J/0/
      READ (8,100,END = 50)  I, SCORE
100   FORMAT (5X,151I1,F2.0,1X)
      J = J + 1
      RETURN
50    NRD = J
      RETURN
       END
```

LIST OF REFERENCES

1. Lubin, A. and Osborn, H. G., " A Theory of Pattern Analysis for the Prediction of a Quantitative Criterion", <u>Psychometrica</u>, v. 22, p. 63-73, 1957.

2. Weitzman, R. A., <u>Pattern Analysis: Method and Applications</u>, Technical Report NPS55WZ73091A, Naval Postgraduate School.

3. Polce, E. F., Jr., <u>Development of a Pattern-analysis Technique for Use in the Selection of Predictors</u>, Master's thesis, Naval Postgraduate School.

4. Bauer, H. R., <u>Introduction to ALGOL-W Programming</u>, Stanford University, July 1969.

INITIAL DISTRIBUTION LIST

1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	2
4.	Assoc. Professor R. A. Weitzman, Code 55Wz Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
5.	Assoc. Professor J. K. Arima, Code 55Aa Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	LCDR J. Q. Jackson, Canadian Forces 5731 Atkins Street Ottawa, Ontario, CANADA	3

Thesis 166111
J216 Jackson
c.1 A programme for the
tailored selection of
response patterns.